MANUAL DE INSTALACIÓN

Versión: 4.0.0/2-ago-2013

1. Introducción

1.1 Objeto

En este documento se describe la instalación simplificada de los componentes necesarios y de la aplicación OpenIrekia en un servidor Ubuntu Linux 12.04 LTS. La instalación creada tiene todo los componentes necesarios para el desarrollo con el sistema.

1.2 Pre-requisitos

Para poder seguir el manual es necesario disponer de:

- Servidor con sistema operativo Ubuntu Server 12.04 LTS ya instalado y actualizado y acceso como root (o sudo)
- Conexión a Internet

1.3 Público

Para realizar la instalación es necesario tener conocimientos de administración de un sistema Linux/Unix.

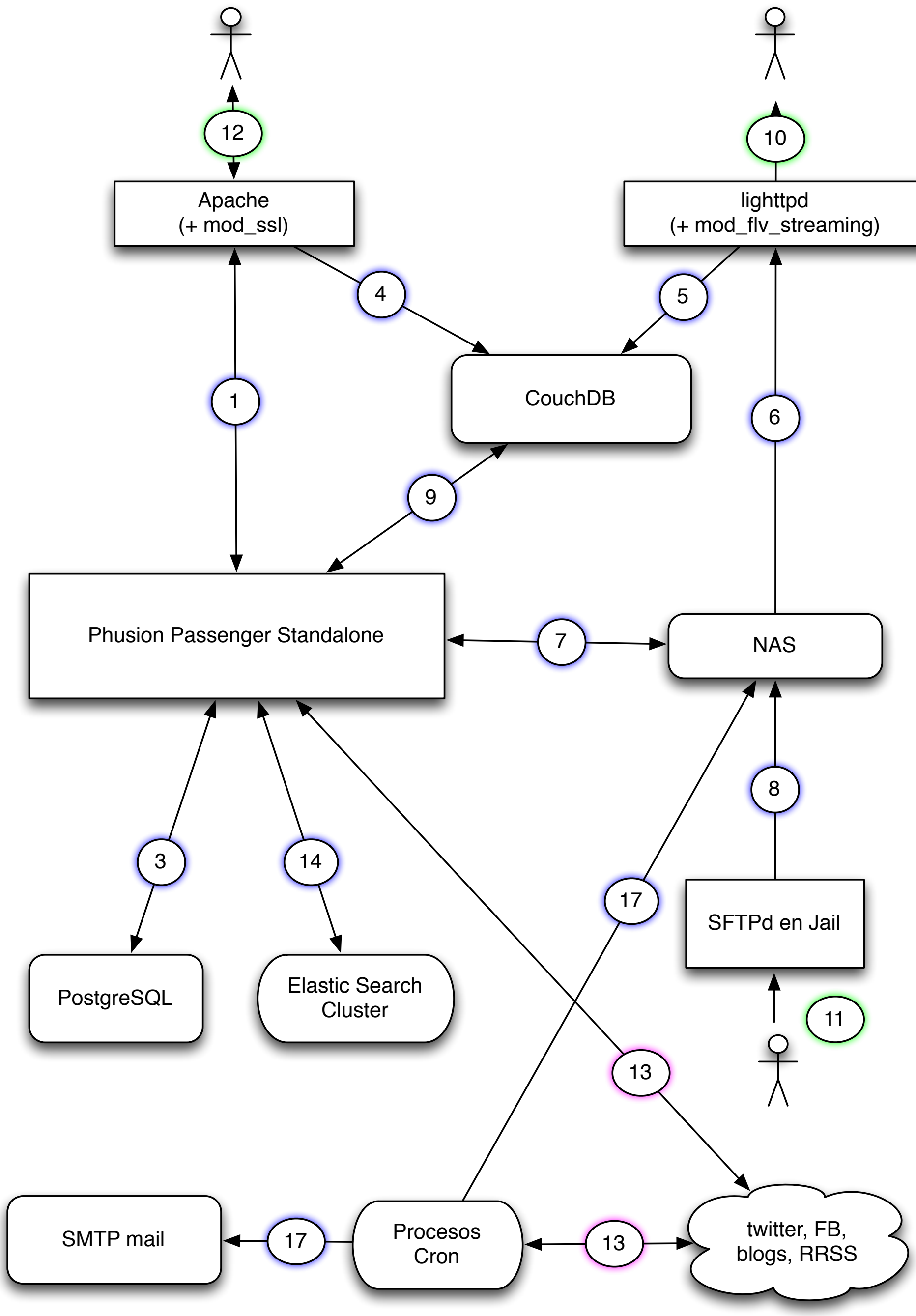
1.4. Licencia

El Gobierno Vasco pone a disposición de usuarios, desarrolladores y comunidad en general la aplicación denominada "OpenIrekia – Gobierno Abierto" bajo la Licencia Pública de la Unión Europea "European Union Public Licence – EUPL". Esta licencia, desarrollada en el seno de la Unión Europea, nació con la intención de ser la licencia bajo la cuál se liberasen los programas y aplicaciones desarrolladas por la Administración Pública y con la característica específica de ser compatible con otras licencias denominadas libres, como la GNU General Public License (GNU/GPL). Estas características dotan, a las aplicaciones así liberadas, de mayor seguridad jurídica y fomentan la interoperabilidad de los servicios de la Administración Electrónica.

The European Union Public Licence http://www.osor.eu/eupl
EUPL v.1.1 - Preámbulo
EUPL v.1.1 – Licencia

2. Estructura y componentes

En la siguiente figura se muestran los principales componentes del sistema en un instalación típica. En una instalación de desarrollo no sería necesario usar NAS externo para el contenido multimedia y acceso restringido para colaboradores por SFTP en jail.



OpenIrekiai

1. El servidor Apache con VirtualHost y configuración SSL, proxy hasta Passenger.

2. -

3. La aplicación usa conexión vía socket local hasta la base de datos postgres
   Protocolo de conexión: socket puerto 5432

4. Los logs de acceso por HTTP y HTTPS al Apache se registran en la base de datos CouchDB
   Protocolo de conexión: REST vía HTTP a puerto 8984

5. Los logs de acceso por HTTP para el contenido multimedia (incluido el streaming vía mod_flv_streaming) se registran en la base de datos CouchDB.
   Protocolo de conexión: REST vía HTTP a puerto 8984

6. -

7. Las aplicaciones crean y trasladan los directorios necesarios y preparan los vídeos en .flv para streaming
   Ficheros de configuración relevantes: /etc/fstab

8. El servidor de SFTP permite deposita el contenido multimedia

9. El módulo de estadística en tiempo real accede a las bases de datos en CouchDB
   Protocolo de conexión: REST vía HTTP a puerto 8984

10. Los usuarios finales acceden a los contenidos multimedia vía navegador web o reproductor flash
    Protocolo de conexión: HTTP

11. Los colaboradores suben contenido pesado (video y fotos) vía SFTP

12. Los usuarios finales y los internos usuarios autorizados acceden a los servicios web
    Protocolos de conexión: HTTP y HTTPS

13. Las aplicaciones usan servicios externos
    Protocolos de conexión: HTTP y HTTPS

14. Las aplicaciones mantienen el índice de búsqueda el texto completo en el servidor ElasticSearch
    Protocolo de conexión: HTTP al puerto 9200

Nota: OpenIrekia gestiona los procesos de streaming en directo pero no incluye servidor de streaming en directo. Actualmente se usa un servicio por parte de los servicios informáticos del gobierno basado en Wowza Media Server y protocolo RTMP.

3. Preparación del servidor

Una vez instalado el sistema operativo básico confirmar la conexión a Internet y seguir los siguientes pasos. Durante la instalación cuando sea posible se usan paquetes de la distribución. Cuando algún componente de software no esta disponible o no esta en la versión necesaria se compila e instala manualmente.

3.1 Paquetes básicos de Ubuntu

Instalar los siguientes paquetes de la distribución:

```
sudo apt-get -y install build-essential libexpat1-dev libxslt1-dev
sudo apt-get -y install lighttpd apache2 curl lynx vim libcurl4-openssl-dev
sudo apt-get -y install libmagick++-dev imagemagick flvmeta qrencode
sudo apt-get -y install postgresql-8.4 postgresql-server-dev-8.4
```

Permitir conexiones locales al servidor PostgreSQL:

```
sudo sed -e \
       's|127.0.0.1/32          md5|127.0.0.1/32          trust|' \
       -i /etc/postgresql/8.4/main/pg_hba.conf

sudo service postgresql restart
sudo -u postgres createlang plpgsql template1
```

3.2 Entorno RVM

```
curl -L https://get.rvm.io | sudo bash -s stable
sudo usermod -a -G rvm `whoami`
```

Con el mismo usuario normal (después de re-login):

```
rvm install 1.8.7

rvm use @global
gem uninstall rake
gem install -v=0.8.7 rake
```

3.3 Usuarios

```
sudo groupadd -g 95 rails
sudo useradd -m -G rails -s /bin/bash openirekia
sudo -u postgres createuser -S -d -R openirekia
sudo mkdir -p /web
sudo chown openirekia:rails /web
usermod -a -G rvm openirekia
```

3.4 Componentes adicionales

3.4.1 ffmpeg

Descargar e instalar las librerías necesarias:

```
sudo apt-get -y install lame faac faad x264
```

Recrear el grupo de paquete de ffmpeg con más encoders que en la configuracón estándar de Ubuntu:

```
cd && mkdir ffmpeg && cd ffmpeg
apt-get source ffmpeg
sudo apt-get build-dep ffmpeg
dpkg-source -x libav_0.8.6-0ubuntu0.12.04.1.dsc
cd libav-0.8.6/
DEB_BUILD_OPTIONS="--enable-gpl --enable-nonfree --enable-pthreads \
                   --enable-libfaac --enable-libfaad --enable-libmp3lame \
                   --enable-libx264" fakeroot debian/rules binary
cd ..
sudo dpkg -i *deb
```

3.4.2 stream segmenter

Descargar e instalar el stream segmenter y las dependencias necesarias:

```
sudo apt-get -y install libfaac-dev libfaad-dev libmp3lame-dev libx264-dev
wget http://www.efaber.net/ogov/segmenter_p201308.tar.gz
tar --no-same-permissions -zxvof segmenter_p201308.tar.gz
cd segmenter/
make && sudo make install
```

Añadir los headers correctos para el streaming por http en la configuración del servidor Apache en/etc/apache2/mods-available/mime.conf

```
AddType application/x-mpegURL .m3u8
AddType video/MP2T .ts
```

Añadir los headers correctos para el streaming por http en la configuración del servidor lighttpd en /etc/lighttpd/ o revisando /etc/mime.types

```
".m3u8"   =>  "application/application/x-mpegURL",
".ts"     =>  "video/MP2T",
```

3.5 Proxy Apache

Crear hosts virtuales para las conexiones por http y https y redirigirlos al puerto interno en cual esta iniciada la aplicación, por ejemplo:

```
NameVirtualHost -la-ip-externa-:80
<VirtualHost -la-ip-externa-:80>
 ServerName www.irekia.example.com
 ServerAdmin irekiaadmin@example.com

 RewriteEngine On
 RewriteCond %{HTTPS} !=on
 RewriteRule ^/sadmin(.*) https://%{SERVER_NAME}/sadmin$1 [R,L]
 RewriteRule ^/admin(.*) https://%{SERVER_NAME}/admin$1 [R,L]

 ProxyRequests Off
 ProxyPass / http://localhost:3000/ retry=5
 ProxyPassReverse / http://localhost:3000/
</VirtualHost>
```

Configurar el acceso por HTTPS en ssl.conf prestando atención a los siguientes parámetros:

```
<VirtualHost -ip-interna-:443>
  ServerName -nombre-publico-
  SSLEngine on
  SSLProtocol all -SSLv2
  SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
  SSLCertificateFile -path-al.cer-
  SSLCertificateKeyFile -path-al.key-

  SetEnvIf User-Agent ".*MSIE.*" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

  ProxyRequests Off
  ProxyPass / http://localhost:3000/ retry=5
  ProxyPassReverse / http://localhost:3000/
  RequestHeader set X_FORWARDED_PROTO "https"
</VirtualHost>
```

3.6 Servidor ElasticSearch

Instalar ElasticSearch versión 0.90.2 de http://www.elasticsearch.org/download/

```
sudo apt-get -y install openjdk-7-jre-headless
wget https://download.elasticsearch.org/elasticsearch/elasticsearch/ \
                                                elasticsearch-0.90.2.deb
sudo dpkg -i elasticsearch-0.90.2.deb
```

3.7 Componentes para estadística en tiempo real

Instalar CouchDB:

```
sudo apt-get -y install couchdb
sudo service couchdb start
```

Activar el logging desde los servidores Apache y lighttpd al CouchDB:

Instalar el log-reader:

```
sudo apt-get -y install python-dev python-simplejson
sudo apt-get -y install python-httplib2 python-couchdb
wget http://www.efaber.net/ogov/log_reader_git.tar.gz
tar -xzvf log_reader.tar_git.gz
cd log_reader_git
sudo python setup.py install
```

Crear la base de datos para logs:

```
curl -X PUT http://localhost:5984/ilog3
curl -X PUT http://localhost:5984/wlog4
```

Configurar el logging en el Apache añadiendo en las secciones VirtualHost:

```
    CustomLog "|python /usr/local/bin/a2c.py irekia" combined
y   CustomLog "|python /usr/local/bin/a2c.py irekia_https" combined
```

Configurar el lighttpd en /etc/lighttpd/lighttpd.conf:

```
accesslog.filename = "|python /usr/local/bin/a2c.py video"
```

4. OpenIrekia

4.2 Aplicación OpenIrekia

Descargar el código fuente de la aplicación (como usuario openirekia):

```
cd /web
wget http://bideoak2.euskadi.net/openirekia/ogov/v4/OpenIrekia-4.0.0.tar.gz
tar -zxvf OpenIrekia-4.0.0.tar.gz
```

Para incluir contenidos multimedia de ejemplo:

```
cd openirekia4/public/
mkdir mm && cd mm
wget http://bideoak2.euskadi.net/openirekia/ogov/v4/OpenIrekia-mmedia-4.0.0.tar.gz
tar -zxvf OpenIrekia-mmedia-4.0.0.tar.gz
rm OpenIrekia-mmedia-4.0.0.tar.gz
cd /web/openirekia4
```

Instalar los gems necesarios:

```
rvm gemset create openirekia4
rvm use 1.8.7@openirekia4
wget http://www.efaber.net/ogov/openirekia4.gems
rvm gemset import openirekia4.gems
gem uninstall rake -v=10.1.0
gem install passenger
```

Crear la base de datos y cargar datos de ejemplo inicial:

```
rake db:create:all
rake db:schema:load
rake openirekia:load_sample_data

compass compile
rake asset:clean
rake asset:compress
```

Finalmente preparar el servidor de búsqueda:

```
rake elasticsearch:irekia_export_all
```

Iniciar con:

```
script/server
```

o alternativamente para iniciar con Passenger Standalone:

```
passenger start
```

Después de ejecutar rake openirekia:load_sample_data la base de datos contiene varios ejemplos de diferentes tipos de contenido y varios usuarios:

ROLE	EMAIL	CONTRASEÑA
Administrador	admin@example.com	openirekia
Jefe de departamento	jefe_dept@example.com	openirekia
Operador de streaming	operador@example.com	openirekia
Político	politician@example.com	openirekia

4.3 Parámetros de configuración

Hay varios parámetros de configuración en config/environment.rb:

- Obtener Google Maps Api Key desde https://developers.google.com/maps/signup?csw=1 y guardar su valor debajo de <GOOGLE_GEOCODE_KEY>
- Por motivos de segurdad se recomienda cambiar los valores de config.action_controller.session key y el secret
- La configuración del servidor de email esta en ActionMailer::Base.smtp_settings
- Si se desea login con Facebook crear aplicación en https://developers.facebook.com/docs/facebook-login/encouraging-facebook-login/. Después modificar config/facebooker.yml
- Para la integración con Twitter seguir las instrucciones en https://dev.twitter.com/docs/auth/sign-twitter. Después modificar config/twitter.yml
- Los contenidos multimedia se buscan en el path indicado en config/multimedia_directories.yml
- La configuración del servidor ElasticSearch esta en config/elasticsearch_server.rb

4.4 Procesos periódicos

Para el correcto funcionamiento del sistema se ejecutar periódicamente varios procesos. Los propios programas son scripts de rake y bash, se encuenran en batch_processes y se pueden ejecutar a través de cron:

```
# Los periodistas se pueden suscribir a los eventos de uno o varios departamentos.
# Cada vez que se publica un evento, se programan las alertas para los periodistas.
# Este cron envía esas alertas cada hora
7 * * * * sh /web/openirekia4/batch_processes/process_unsent_alerts_for_journalists.sh

# Los responsables de salas de streaming y los operadores de streaming reciben alertas
# sobre los eventos que se van a emitir por streaming. Este cron envía esas alertas
# cada 5 minutos
*/5 * * * * sh /web/openirekia4/batch_processes/process_unsent_alerts_for_staff.sh

# Los vídeos que se suben por SFTP en el directorio multimedia de cada noticia
# deben importarse en la videoteca. Este cron busca nuevos vídeos en las últimas
# noticias publicadas y los importa.
22 * * * * sh /web/openirekia4/batch_processes/include_new_videos_in_webtv.sh

# Las fotos que se suben por SFTP en el directorio multimedia de cada noticia
# deben importarse en la fototeca. Este cron busca nuevos vídeos en las últimas
# noticias publicadas y los importa.
16 12,15,18,21,0 * * * sh
/web/openirekia4/batch_processes/include_new_photos_in_gallery.sh

# Todas las noticias y eventos publicadas en la web son tweeteadas en sendas cuentas
# de Twitter.
*/20 * * * * sh /web/openirekia4/batch_processes/tweet_pending_issues.sh

# En ocasiones los vídeos de las noticias están disponibles antes en un idioma que
# en otro. Este cron comprueba si hay una nueva versión en un nuevo idioma para
# un video ya importado y lo oferta en la videoteca en este nuevo idioma.
43 * * * * sh /web/openirekia4/batch_processes/check_webtv_video_languages.sh

# Todos los martes, se importan los acuerdos de consejo de gobierno publicados en
euskadi.net
30 14,16 * * 2 sh /web/openirekia4/batch_processes/import_consejo_news.sh

# Actualizar los contadores de mpg y mp3
13 3 * * * sh /web/openirekia4/batch_processes/update_file_system_stats.sh

# Búsqueda tweets que mencionan Irekia. Si se encuentran, se intenta identificar
# el contenido al que referencian (noticia, evento, propuesta, etc) y se importa en
# el mismo como un comentario. Ese comentario debe ser posteriormente aprobado por un
# administrador como cualquier otro comentario
56 */2 * * * sh /web/openirekia4/batch_processes/import_tweets_as_comments.sh
```